

УДК 004.65, 004.42

С.К.Бурков (3 курс, каф. ИУС), Т.А.Вишневская, ст. препод.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ADO.NET В РАЗРАБОТКЕ БИЗНЕС-ПРИЛОЖЕНИЙ

На данный момент на рынке средств разработки программного обеспечения предлагается множество различных технологий работы с данными, в том числе и технологии корпорации Microsoft. Таким образом, перед разработчиками приложений для работы с базами данных (БД) возникает вопрос: какую из предлагаемых технологий использовать с точки зрения скорости разработки программного продукта, быстродействия, эффективности и надежности создаваемого приложения, удобства работы конечного пользователя с приложением.

В качестве объектов исследования были выбраны две последние технологии доступа к данным Microsoft – ActiveX Data Objects (ADO) и ActiveX Data Objects .NET (ADO.NET).

Целью данного исследования было определение достоинств и недостатков данных технологий. Необходимо было оценить их с точки зрения следующих критериев: скорости обмена данными с базой данных, возможностей по управлению и манипуляции над данными и предоставляемых библиотек классов элементов управления и работы с данными.

В качестве модельной задачи была выбрана следующая:

- приложение должно корректно работать с иерархично-связанными данными;
- приложение должно предоставлять максимум удобств для пользователя;
- разработка приложения должна вестись с использованием предоставляемых средствами разработки средств автоматизации.

Были написаны приложения, удовлетворяющие данной задаче, с использованием технологий доступа к данным ADO и ADO.NET. Разработка приложений велась в MS Visual Studio 6.0 на языке Visual Basic 6.0 и в MS Visual Studio .NET 2003 на языке C#, соответственно.

В результате исследования были выявлены следующие отличия между этими технологиями:

ADO	ADO.NET
Нет возможности повлиять на обновление данных, т.к. все автоматически генерируемые командные запросы скрыты от программиста	Есть возможность изменить автоматически генерируемую логику изменения и обновления данных
Обновления должны передаваться непосредственно в таблицы БД	Передавать изменения в БД можно с использованием хранимых процедур
Ядро курсоров ADO не умеет собрать метаданные, необходимые для передачи изменений в БД, а предоставить такие данные программно нельзя	Объект <i>DataSet</i> разрешается заполнять результатами вызова хранимой процедуры, запроса к временной таблице, сводными результатами множества запросов или любым другим способом, и все равно остается возможность передать изменения в БД
Ядру курсоров ADO необходимо получать из БД имя исходной таблицы, исходных столбцов, а также сведения о первичном ключе и ограничениях исходной таблицы	Загрузка таких данных может происходить из локального кода

Интерфейсы ADO используется в качестве «посредника» при взаимодействии с БД. ADO передает поставщику данных все ваши вызовы. Поставщик выполняет необходимые действия и возвращает вам результат через библиотеку ADO	У поставщиков данных .NET промежуточный уровень отсутствует. Программа обращается непосредственно к поставщику, и тот взаимодействует с хранилищем данных при помощи низкоуровневых интерфейсов программирования последнего
Пул соединений отсутствует	Присутствует пул соединений (Открытие и закрытие соединений с БД – дорогостоящие операции. Использование пула при разработке многоуровневого приложения позволяет значительно повысить его производительность)
При использовании полей таблиц, основанных на выражении, для корректного отображения данных приходится перевыбирать данные из БД	Присутствует возможность «запрограммировать» поле, основанное на выражении и не делать перевыборку данных из БД
Нет возможности программно генерировать временные значения для автоинкрементных полей – при добавлении новых данных идет обязательная выборка следующего значения автоинкремента из БД	Есть возможность программно генерировать временные значения для автоинкрементных полей
При локальном удалении данных в объекте <i>Recordset</i> ADO проверяет, не связаны ли они друг с другом в БД, независимо от того заданы ли отношения между ними или нет, и генерирует исключение ссылочной целостности в случае обнаружения такой связи в БД	Мы можем удалить локальные данные из таблиц и при этом не удалять их в БД
Объекты <i>Recordset</i> модели ADO поддерживают концепцию «текущего ряда» данных	При использовании объекта <i>DataTable</i> всегда доступны все записи

После анализа отличий между технологиями были выявлены достоинства и недостатки технологий ADO.NET перед ADO.

Преимущества ADO.NET:

- при использовании собственной логики обновления можно воспользоваться всеми средствами, предоставляемыми СУБД;
- происхождение данных не влияет на возможность обновления;
- повышенная производительность;
- постоянное подключение к БД не требуется.

Недостатки ADO.NET: использования собственной логики обновления в коде – зеркальное отражение преимуществ ядра курсоров ADO. Собственная логика обновления занимает больше места. Написание такого кода — утомительное занятие, отнимающее много времени.

В результате проведенного исследования по сравнению технологий доступа к данным ADO и ADO.NET были выработаны следующие рекомендации: при написании достаточно простых приложений, ориентированных только на выборку данных, достаточно воспользоваться функциональностью технологии ADO. Если же приложение будет активно обрабатывать данные и модифицировать их, то его лучше и проще написать с использованием технологии ADO.NET.

ЛИТЕРАТУРА:

1. Дженнингс Р. Руководство разработчика баз данных на Visual Basic 6.0. СПб.: «Вильямс», 2000. с. 976.
2. Сеппа Д. Microsoft ADO.NET. М.: «Русская редакция», 2003. с. 640.